Scientific Visualization Project

Open Enrollment Plan Selection - 2020 Analysis using Observable & Vegalite

Created By: Lavanya Rao

**Abstract:**

Charts have been one of the most popular ways of displaying data and inviting further exploration on data. Charts can be used in situations where a simple table may not be able to efficiently convey/demonstrate the data. There are multiple tools available to create charts. In this project, I have used the vegalite language on an Observable notebook to create charts and visualize the healthcare Plan data from the CMS website.

**Introduction:**

This project on Scientific Visualization is created using the Observable HQ with Vegalite. Observable HQ is very similar to javascript, but at the same time offers some changed behaviours than javascript. Observable is built on the native language of the web and hence is more familiar and you can use libraries such as D3, tensorflow etc.

For rapid charting and fast exploration of data in observable HQ, Vegalite is recommended. Hence the project uses the basic features of chart creation using vegalite on an observable notebook to analyze the Health Insurance Exchange’s Open Enrollment data for 2020. The data is derived from the Center of Medicare and Medicaid services’ 2020 OEP.

The state-level PUF includes total health plan selections in all 50 states plus the District of Columbia. The PUF provides state-level data on metrics such as average monthly premium, financial assistance, age, gender, metal level, self-reported race and ethnicity, rural location, household income as a percent of the federal poverty level (FPL), and plan switching behavior among consumers with a plan selection.

**Project Details:**

**Introduction to Observable:**

* Each cell in an observable notebook is a separate script that runs independently.
* The cells run in topological order, so you can arrange the cells as you want the order of the code.
* Cells rerun automatically when the value of any other referenced cell changes.
* A cell can be defined with a promise and if it is done, the referencing cell won’t run until the value resolves.
* Named cells are declarations and not assignments.
* Statements need curly braces and return.
* Cells can be imported from other notebooks.

**Introduction to Vegalite:** [**Vega-Lite**](https://vega.github.io/vega-lite) is a declarative language for interactive data visualization. Vega-Lite offers a powerful and concise visualization grammar for quickly building a wide range of statistical graphics. It is declarative in the sense that you can define how you want the visualization to appear. You can provide high level specifications of what you want the visualization to include rather than specifying how to include it. This feature makes it a very powerful tool as the analyst/developers can focus more on the visualization aspect of it, rather than the technical details.

The data in Observable can be loaded into the notebook in different ways.

1. Inline- This means the data can be declared as arrays and embedded in the code in the notebook.
2. File- For medium amounts of data, the data file can be imported and attached to the notebook as an attachment. The attachment can then be accessed in the code in the cell. This has been used in the current project for loading the data file.
3. APIs- The data can also be fetched from a remote server via API calls.
4. Databases- The data can also be accessed via an observable database client for accessing SQL databases.

**2020 Marketplace Open Enrollment Period**

For the 38 states using HealthCare.gov (HC.gov), the Health Insurance Exchange reporting period reflects plan selection and Exchange activity from November 1, 2019 to December 21, 2019.

The state-level PUF includes total health plan selections in all 50 states plus the District of Columbia. The PUF provides state-level data on metrics such as average monthly premium, financial assistance, age, gender, metal level, self-reported race and ethnicity, rural location, household income as a percent of the federal poverty level (FPL), and plan switching behavior among consumers with a plan selection. In addition, the state-level PUF includes data on dental plan selections and Basic Health Plan (BHP) enrollments. Certain data elements are only available for the 38 HC.gov states in 2020.

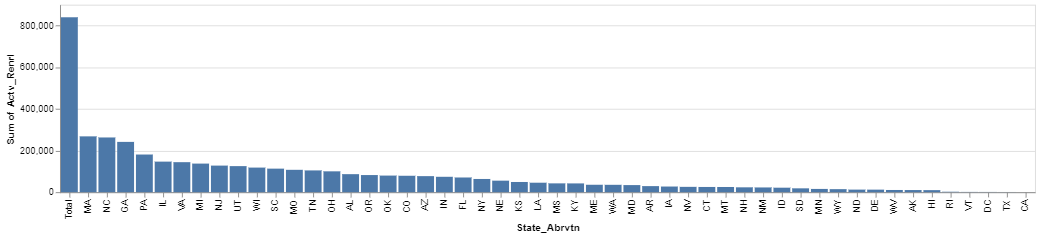
**Data Cleaning Process:** The data used in the project was downloaded from the public use files provided by CMS(Center of Medicare and Medicaid services). The data had to be cleaned before making it available for chart creation. The data cleaning was done using notepad++. Most of the numeric data was embedded in double quotes , with a comma to separate the least significant and most significant bits. These had to be changed into numeric values so as to use them in computation. This was done, by using the below regular expression replace strategy on the csv file.

Find: (\".\*?),(.\*?\")

Replace: \1\2

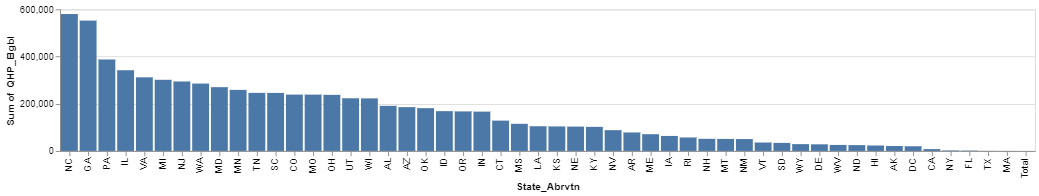
**The Charting Process:**

For creating the bar chart , Vegalite script is used. The script uses the bar chart and the encoding scheme for the x and y axis. The chart 1 shows for each State the total number of active renewals and the chart 2 shows for each state the QHP eligibility. The chart is also sorted according to aggregated y values using the sort keyword.



(fig.1)

Number of Active Enrollments per State



(fig.2)

Sum of QHP Eligibility per State

**Link to the Observable Notebook: (ctrl+click on the below text to view the link and then click on the link to view the Observable notebook)**

[**OEPPlanAssignment / Lavanya Rao / Observable (observablehq.com)**](https://observablehq.com/@lrao3/oep2020plans)

**Conclusion:**

Observable introduces the notebook paradigm to JavaScript projects. People are already familiar with the reactive ways in which javascript can be used and hence observable provides an easier way to use the javascript. Not only that, Observable does this all with declarative syntax, which makes it easier to understand and focus on the business aspects rather than the technical details. Observable integrates D3 , Vegalite and other programming languages very well and is extensible.

**References:**

* [Home - Centers for Medicare & Medicaid Services | CMS](https://www.cms.gov/)
* [Observable (observablehq.com)](https://observablehq.com/)
* DataSource: [2020 Marketplace Open Enrollment Period Public Use Files | CMS](https://www.cms.gov/Research-Statistics-Data-and-Systems/Statistics-Trends-and-Reports/Marketplace-Products/2020-Marketplace-Open-Enrollment-Period-Public-Use-Files)